

Corso di Algoritmi e Strutture Dati (IN110) – Prof. Marco Liverani – a.a. 2024/2025

Prima prova di esonero – 15 novembre 2024

La prova dura tre ore, durante le quali non è possibile allontanarsi dall'aula, se non dopo aver consegnato l'elaborato scritto. Per superare la prova di esonero e poter sostenere la successiva è necessario ottenere almeno 15 punti. È possibile utilizzare libri e appunti personali, senza scambiarli con altri studenti. I compiti che presenteranno evidenti ed anomale "similitudini" saranno annullati.

Deve essere consegnata solo la "bella copia" del compito scritto; su ciascun foglio deve essere riportato il nome, il cognome e il numero di matricola (o un altro codice identificativo di fantasia) dello studente.

Esercizio n. 1 (max 10 punti)

Risolvere il seguente problema proponendo una **pseudo-codifica dell'algoritmo**, il **diagramma di flusso** ed infine la **codifica in linguaggio C** del programma che implementa l'algoritmo stesso.

Letti in input tre numeri interi n , h e k , con $n > 0$ e $0 < h < k < 100$, costruire un array A con n interi casuali compresi tra h e k (estremi inclusi). Stampare in output l'array A . Stampare l'elemento di A che è presente con maggiore frequenza nell'array.

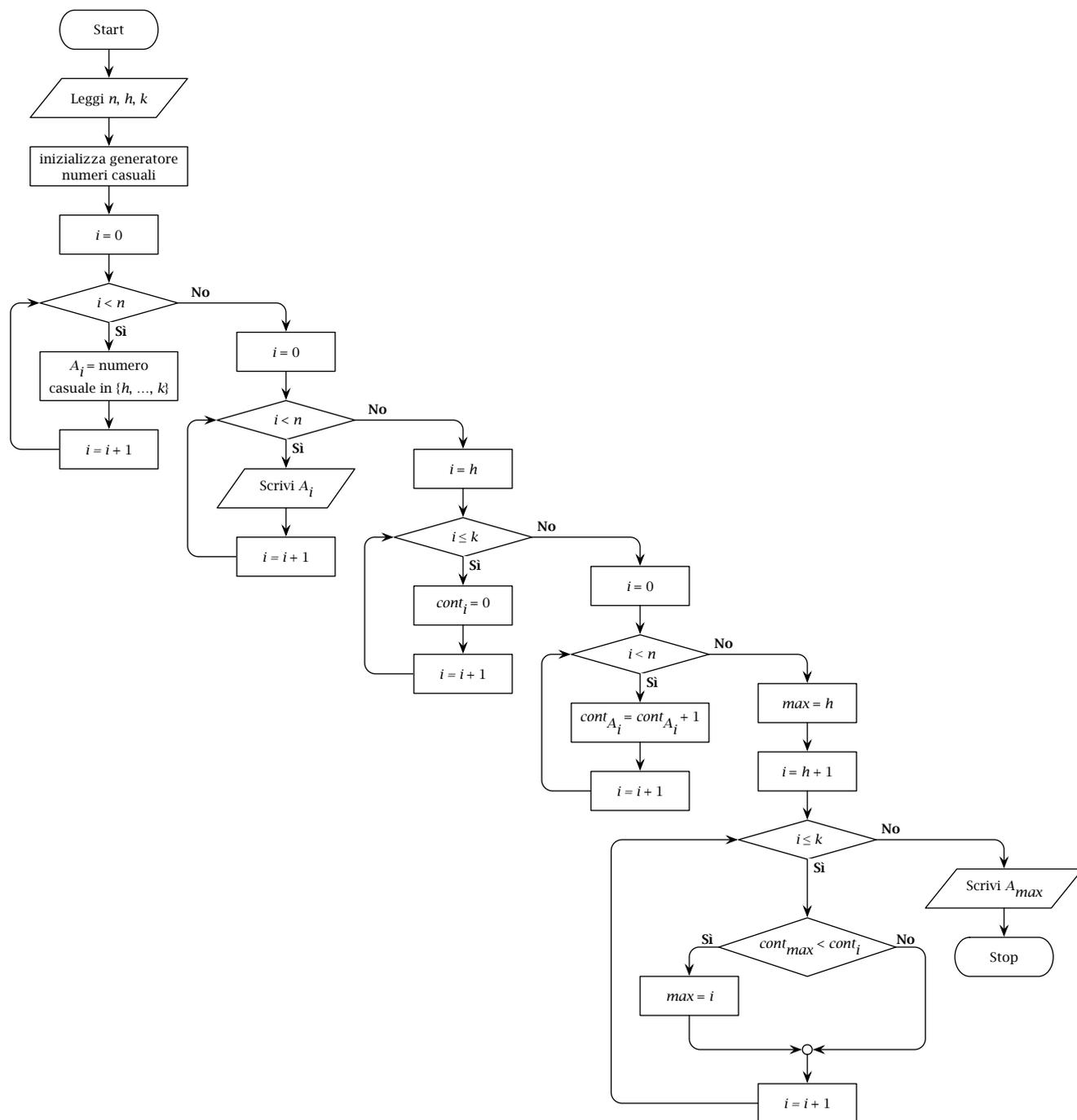
Esempio Siano $n = 11$ e $h = 3$, $k = 81$. Sia inoltre $A = (3, 19, 4, 52, 81, 3, 7, 4, 4, 52, 7)$. L'elemento che è presente con maggiore frequenza in A (tre volte) è 4.

Soluzione

Pseudo-codifica dell'algoritmo

- 1: leggi n , h e k
- 2: inizializza il generatore dei numeri casuali
- 3: per $i = 0, 1, \dots, n - 1$ ripeti:
- 4: $A_i =$ numero casuale in $\{h, \dots, k\}$
- 5: fine-ciclo
- 6: per $i = 0, 1, \dots, n - 1$ ripeti:
- 7: scrivi A_i
- 8: fine-ciclo
- 9: per $i = h, h + 1, \dots, k$ ripeti:
- 10: $cont_i = 0$
- 11: fine-ciclo
- 12: per $i = 0, 1, \dots, n - 1$ ripeti:
- 13: $cont_{A_i} = cont_{A_i} + 1$
- 14: fine-ciclo
- 15: $max = h$
- 16: per $i = h + 1, \dots, k$ ripeti:
- 17: se $cont_{max} < cont_i$ allora
- 18: $max = i$
- 19: fine-condizione
- 20: fine-ciclo
- 21: **return** max

Diagramma di flusso



Codifica in linguaggio C

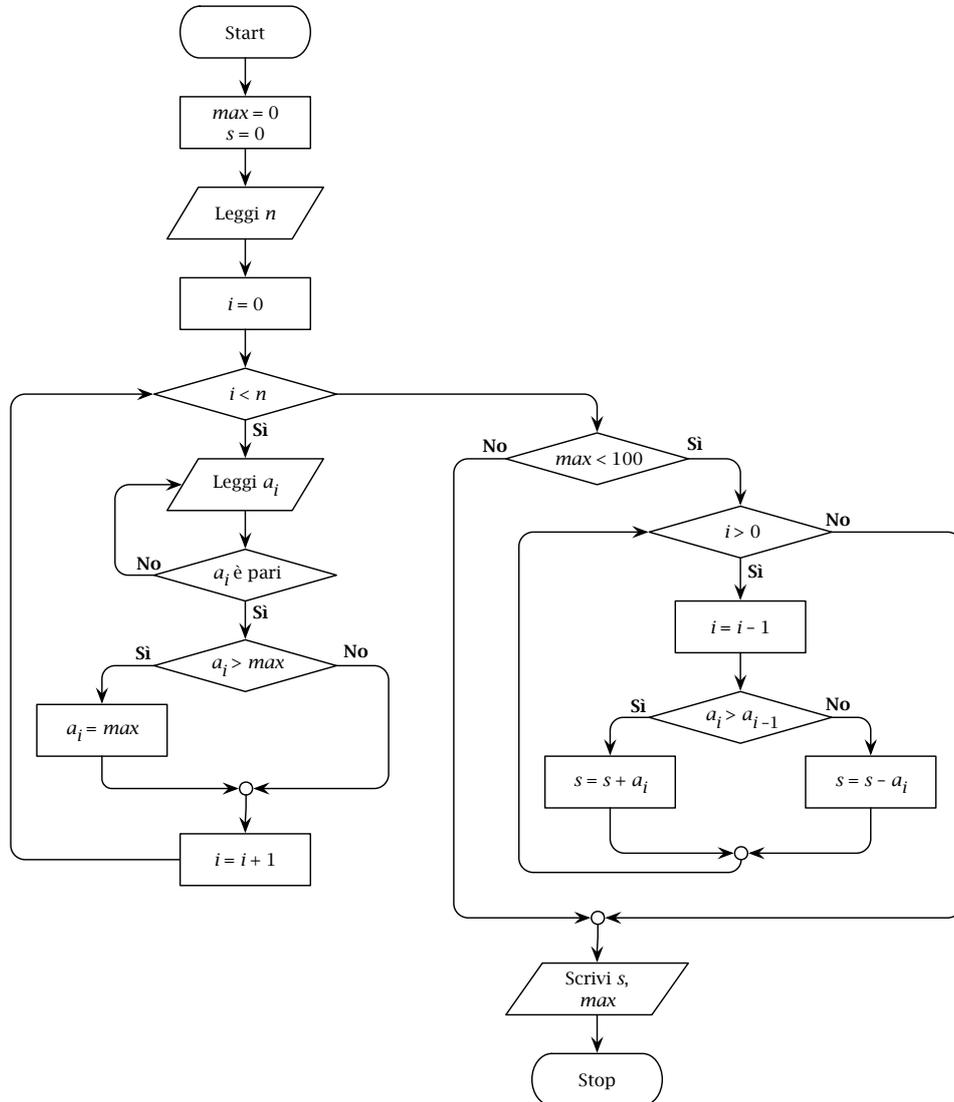
```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 int arrayCasuale(int x[], int h, int k) {
6     int n, i;
7     srand((unsigned)time(NULL));
8     printf("Numero di elementi: ");
9     scanf("%d", &n);
10    for (i=0; i<n; i++)
11        x[i] = rand() % (k - h + 1) + h;
12    return(n);
13 }
14
15 void scriviArray(int x[], int n) {
16     for (int i=0; i<n; i++)
17         printf("%d ", x[i]);
18     printf("\n");
19     return;
20 }
21
22 int maxFrequenza(int A[], int n, int h, int k) {
23     int max, cont[100], i;
24     for (i=h; i<=k; i++)
25         cont[i] = 0;
26     for (i=0; i<n; i++)
27         cont[A[i]]++;
28     max = h;
29     for (i=h+1; i<=k; i++)
30         if (cont[max] < cont[i])
31             max = i;
32     return(max);
33 }
34
35 int main(void) {
36     int A[1000], n, m, h, k;
37     printf("Estremi dell'intervallo: ");
38     scanf("%d %d", &h, &k);
39     n = arrayCasuale(A, h, k);
40     scriviArray(A, n);
41     m = maxFrequenza(A, n, h, k);
42     printf("Elemento con la massima frequenza: %d\n", m);
43     return(0);
44 }
```

Esercizio n. 2 (max 8 punti)

Scrivere il **diagramma di flusso** e la **codifica in linguaggio C** (un programma completo) del seguente algoritmo:

```
1:  $max = 0, s = 0$ 
2: leggi  $n$ 
3: per  $i = 0, 1, \dots, n - 1$  ripeti:
4:   ripeti:
5:     leggi  $a_i$ 
6:     fintanto che  $a_i$  non è pari
7:     se  $a_i > max$  allora
8:        $max = a_i$ 
9:     fine-condizione
10:  fine-ciclo
11: se  $max < 100$  allora
12:  fintanto che  $i > 0$  ripeti:
13:     $i = i - 1$ 
14:    se  $a_i > a_{i-1}$  allora
15:       $s = s + a_i$ 
16:    altrimenti
17:       $s = s - a_i$ 
18:    fine-condizione
19:  fine-ciclo
20: fine-condizione
21: scrivi  $s, max$ 
```

Diagramma di flusso



Codifica in linguaggio C

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 int main(void) {
6     int max=0, s=0, n, i, a[100];
7     scanf("%d", &n);
8     for (i=0; i<n; i++) {
9         do {
10            scanf("%d", &a[i]);
11        } while (a[i]%2 != 0);
12        if (a[i] > max)
13            max = a[i];
14    }
15    if (max < 100) {
16        while (i > 0) {
17            i--;
18            if (a[i] > a[i-1])
19                s = s + a[i];
20            else
21                s = s - a[i];
22        }
23    }
24    printf("%d %d\n", s, max);
25    return 0;
26 }
```

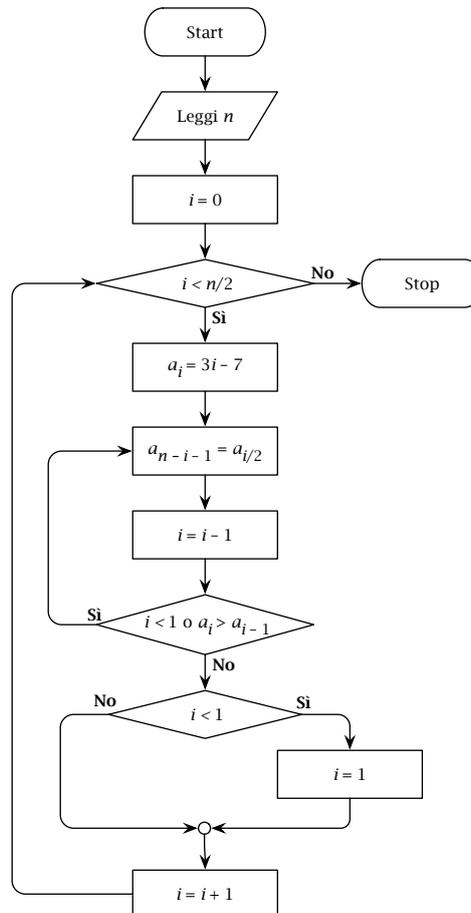
Esercizio n. 3 (max 6 punti)

Utilizzando le regole della programmazione strutturata, disegnare il **diagramma di flusso** della seguente funzione codificata in linguaggio C.

Nota: la funzione riporta istruzioni sintatticamente corrette, ma complessivamente prive di significato.

```
1 void funzione(float a[100], int n) {  
2   int i;  
3   scanf("%d", &n);  
4   for (i=0; i<n/2; i++) {  
5     a[i] = i*3-7;  
6     do {  
7       a[n-i-1] = a[i]/2;  
8       i = i-1;  
9     } while (i < 1 || a[i] > a[i-1]);  
10    if (i < 1) {  
11      i = 1;  
12    }  
13  }  
14  return;  
15 }
```

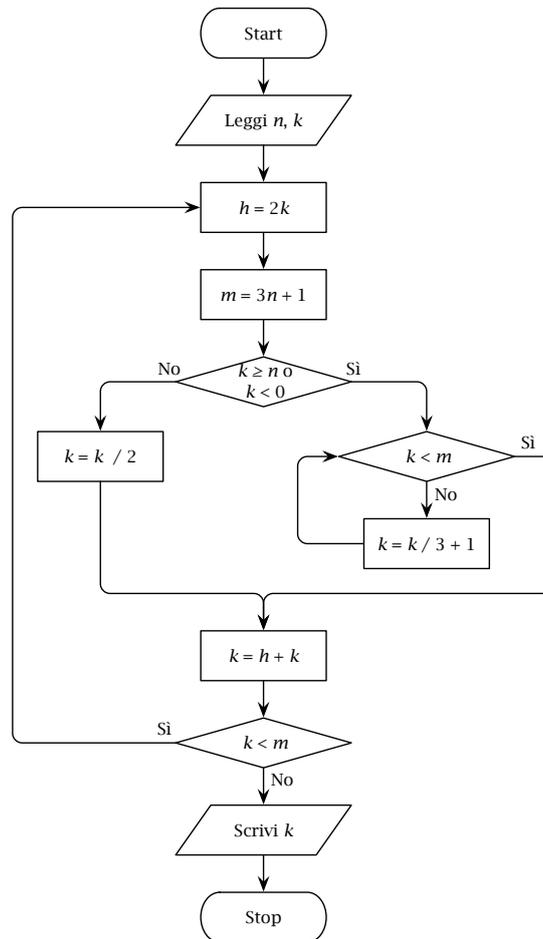
Soluzione



Esercizio n. 4 (max 6 punti)

Codificare in linguaggio C un **programma completo** che implementi l'algoritmo descritto dal seguente diagramma di flusso. Nella codifica porre attenzione anche alla definizione delle variabili e degli array.

Nota: il diagramma di flusso è formalmente corretto, anche se rappresenta un algoritmo privo di significato.



Soluzione

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main(void) {
5     int n, m, h, k;
6     scanf("%d %d", &n, &k);
7     do {
8         h = 2*k;
9         m = 3*n + 1;
10        if (k >= n || k < 0)
11            while (k >= m)
12                k = k/3 + 1;
13        else
14            k = k/2;
15        k = h + k;
16    } while (k < m);
17    printf("%d", k);
18    return 0;
19 }
```